

# Summary of Best Efficient Algorithms on Cognitive Radio Cooperative Spectrum Sensing to Improve QoS

Mohsen Akbari, Ahmed Wasif Reza, Kamarul Ariffin Noordin, Mohsen Riahi Manesh  
and Mohammad Nour Hindia

*Faculty of Engineering, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, 50603  
Kuala Lumpur, Malaysia  
[Mohsen.akbari61@yahoo.com](mailto:Mohsen.akbari61@yahoo.com)*

**Abstract:** Cooperative spectrum sensing (CSS) in cognitive radio (CR) has been widely investigated to be considered as a spectrum scanning mechanism that allows secondary users or cognitive radio users (SUs) to use detected spectrum holes caused by primary users (PUs) absence. This paper focuses on optimality of analytical study on the common soft fusion (SDF) CSS based on different iterative algorithms which confirm low total probability of error and high probability of detection in details. In fact, all steps of genetic algorithm (GA), particle swarm optimization (PSO) and imperialistic competitive algorithm (ICA) will be well mentioned in details and investigated on cognitive radio cooperative spectrum sensing (CRCSS) method. Then, the performance of CRCSS employing GA, POS and ICA based scheme is analyzed in MATLAB simulation to show superiority of these schemes over other conventional schemes in terms of detection and error performance with very less complexity. In addition, the ICA-based scheme also indicates promising convergence and time running performance as compared to other schemes.

**Keywords:** Cognitive radio, cooperative spectrum sensing, soft decision fusion, ICA, PSO, GA.

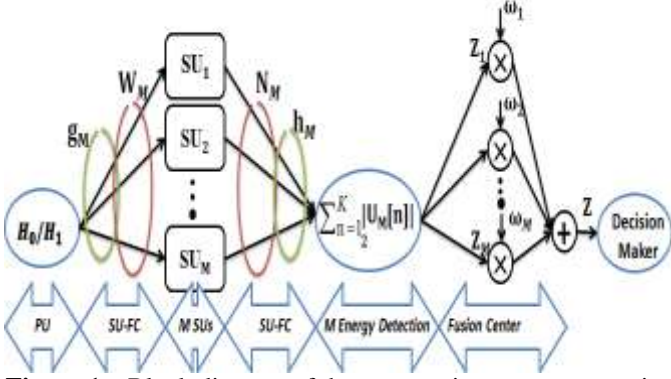
## 1. Introduction

Radio spectrum is a precious resource and characterized by fixed allocation policy. However, mostly allocated spectrum is underutilized by licensed users [1]. Conversely, the rapid development of ubiquitous wireless technologies increases the demand for radio spectrum. Federal Communications Commission (FCC) used the Static Spectrum Allocation (SSA) scheme to allocate spectrum bands to users exclusively since licensed users do not occupy radio spectrum or even at least whole of it all the time. As a solution of the spectrum inefficiency problem, cognitive radio is an exciting and emerging technology, which has attracted a great deal of attention in recent years to enhance the utilization of limited resources [1]. In cognitive radio network, licensed or primary users (PU) coexisted with the unlicensed or secondary users (SU) in the same frequency band to achieve better spectrum utilization. [2]. Also key technology that can help mitigate the scarcity of spectrum is CR[3], [4]. In other word, SUs can opportunistically access the licensed spectrum without causing interferences to the PUs. By using this access strategy, the spectrum resources can be assured enhance of spectrum

efficiency and thus, significant increases of the number of users using wireless services which can greatly resolve spectrum scarcity problem. However, detection performance may affect by shadowing effect and the hidden terminal problem and SU may not detect the activity of the PU within the short interval of sensing period [5]. Thus, to solve this issues CSS was proposed by [6, 7], it is based on keep monitoring the spectrum periodically to overcome hidden terminal and shadowing problems for minimize interference. The well-established local sensing mechanism for spectrum sensing in cognitive radio networks is called as energy detection. It does not require the prior knowledge about the signal of PU. Also, it has short sensing time as well as represents a simple technique, low implementation cost and compatibility with legacy primary systems. The energy detectors are low complex as measuring received signal's power to check the existence of PU with unknown power strength, waveform structure and frequency location. The results which are collected from energy detectors will be forwarded to a Fusion Center (FC) where the global decision on the existence of PU will be taken based on two methods namely soft decision fusion (SDF) and hard decision fusion (HDF) [8-11]. Also in [12] SDF-based linear CSS methods were applied to find the ideal weights which had been executed based different conventional approaches were implemented at the FC to reduce probability of miss detection ( $P_m$ ), and probability of error ( $P_e$ ). To the best author's understanding, iterative algorithms overcome conventional models in CRCSS issue (PSO is the advanced method till now). This research considers imperialistic competitive algorithm (ICA) based on SDF method and is performed at the FC to reduce global  $P_e$ , increase probability of correct detection ( $P_d$ ) and is compared with other iterative methods, e.g., (PSO and GA). In addition, it has been shown that ICA-based method provides better convergence performance and lower complexity than other existing iterative SDF-based schemes.

## 2. System Model

SDF-based CSS is applied to enhance detection reliability SUs when local CSS is used, thus, local spectrum sensing can be realized by means of energy detection without any prior knowledge of the PU signal [14]. Fig.1 demonstrates the centralized CSS SDF-based scenario where M SUs (as relays) send their measurements on the activity of PU to FC to make decision on the presence of PU [10].



**Figure 1.** Block diagram of the cooperative spectrum sensing

Fig. 1 is modeled as channels starting from PUs to FC are assumed to be Rayleigh fading (different gains) and noise is AWGN with different variance in each path. During the entire frame duration, it is assumed that the PU present in different time in channel, thus, the CSS process can be given as binary hypothesis testing:

$$X_i[n] \begin{cases} g_i S[n] + W_i[n], H_1 \\ W_i[n], H_0 \end{cases} \quad (1)$$

where  $X[n]$  is the signal that is received at the CR user,  $S[n]$  is the transmitted PU signal,  $g$  refers to the gain of the sensing channel,  $W[n]$  is the zero-mean additive white Gaussian noise (AWGN),  $H_0$  and  $H_1$  are the hypothesis of the absence and the presence, respectively, of the PU signal in the frequency band of interest.

CRNs are mathematically modeled under two main criteria: Mini-Max [17] and Neyman-Pearson [16] criterion, which are individually defined as follows:

### a) Neyman-Pearson Criteria

Neyman-Pearson criterion was proposed based on the problem of the most efficient tests of statistical hypotheses [15]. This research applied Neyman-Pearson criterion as possibility of minimal interference which is caused by SU to PU in active position.

$$\begin{aligned} P_d &= \{decision = H_1 | H_1\} = P\{Y > \lambda | H_1\} \& \\ P_f &= \{decision = H_1 | H_0\} = P\{Y > \lambda | H_0\} \end{aligned} \quad (2)$$

where  $Y$  is the decision statistic and  $\lambda$  is the decision threshold. The value of  $\lambda$  at the FC is constant based on the fixed known  $P_f$  and the final  $P_d$  and  $P_f$  expressed as [17]:

$$P_f = P(Y > \lambda | H_0) = Q\left(\frac{\lambda - E(Y|H_0)}{\sqrt{\text{var}(Y|H_0)}}\right) = Q\left(\frac{\lambda - \vec{\omega}^T \vec{\mu}_0}{\sqrt{\vec{\omega}^T \Phi_{H_0} \vec{\omega}}}\right) \quad (3)$$

$$P_d = P(Y > \lambda | H_1) = Q\left(\frac{\lambda - E(Y|H_1)}{\sqrt{\text{var}(Y|H_1)}}\right) = Q\left(\frac{\lambda - \vec{\omega}^T \vec{\mu}_1}{\sqrt{\vec{\omega}^T \Phi_{H_1} \vec{\omega}}}\right) \quad (4)$$

$$P_d(\vec{\omega}) = Q\left(\frac{Q^{-1}(P_f) \sqrt{\vec{\omega}^T \Phi_{H_0} \vec{\omega}} - \vec{\omega}^T \vec{\theta}}{\sqrt{\vec{\omega}^T \Phi_{H_1} \vec{\omega}}}\right) \quad (5)$$

Equation (5) provides a reliable measure of detection performance in SDF-based cooperative sensing for a fixed set of false alarm probabilities.

### b) Mini-Max Criteria

Mini-Max method is able to trade-off between spectrum utilization and interference in PU [10], [18]. In other words, we plan to minimize  $P_e$  and probability of miss-match  $P_m$  which are unwanted in any communication detection task. For simplicity, let us assume that  $P_f$  is the same with  $P_m$ . Based on these definitions, the probability of a miss or miss detection is defined as  $P_m = 1 - P_d = P\{decision = H_0 | H_1\}$ . Total probability of error,  $P_e$  and is simply represented by [16]:

$$P_f = P(Y > \lambda | H_0) = Q\left(\frac{\lambda - E(Y|H_0)}{\sqrt{\text{var}(Y|H_0)}}\right) = Q\left(\frac{\lambda - \vec{\omega}^T \vec{\mu}_0}{\sqrt{\vec{\omega}^T \Phi_{H_0} \vec{\omega}}}\right) \quad (6)$$

It is discernible that maximization of  $P_d(\vec{\omega})$  and minimization of  $P_e(\vec{\omega})$  are very dependent on  $\vec{\omega}$ . As already mention this research will investigate optimal weighting vector for above objective functions with the help of GA, PSO and ICA algorithms with will be explained in detail as below.

### 2.1 GA based cooperative spectrum sensing

GA is one of the stochastic iterative search algorithms that follows natural evolution. It has been used for several engineering applications such as solving complicated non-deterministic problems as well as machine learning. GA is a

population-based method in which each individual in the population evolves to create new individuals that form new populations. This evolutionary procedure iteratively goes on until no improvement on the fitness score is achieved and then the optimal individual (fitness score) is achieved from the last updated population.

In this paper GA-based technique has been proposed, an initial population of  $pops$  possible solutions is generated randomly and each individual is normalized to satisfy the restrictions. Our main goal is to find the optimal set of weighting vector values to maximize detection performance. When the maximum number of generations is reached (predefined level), GA process is ended and the weighted vector values that minimize the probability of error is obtained as the best solution. If we assume that, there are  $M$  SUs and  $Z_1, Z_2, \dots, Z_M$  are the soft decisions of  $SU_1, SU_2, \dots, SU_M$  on the presence of PUs, and  $\bar{\omega}_j$  is the weighting vector of the  $j^{th}$  individual that consists of  $W_1, W_2, W_3, \dots, W_M$ , the fitness value for the  $j$ th individual is defined as follow:

$$f_j = p_e(\bar{\omega}_j) \text{ where } \|\bar{\omega}_j\| = 1 \quad (7)$$

$p_e$  stands for probability of error. The principal functions of the GA are selection, crossover, and mutation. For selection, the idea is to choose the best chromosomes for reproduction through crossover and mutation. The smaller the fitness value (probability of error), the better the solution obtained. In this paper "Roulette Wheel selection" method has been used. The probability of selecting the  $j^{th}$  individual or chromosome  $P_j$  can be written as:

$$P_j = \frac{J_j}{\sum_{j=1}^{pops} f_j} \quad (8)$$

The chromosomes with minimum probability of error value will be transferred to the next generation through elitism operation. After the chosen process is finished, the next step is crossover. The crossover starts with pairing to produce new offspring. A uniform random number generator has been used to select the row numbers of chromosomes as mother ( $ma$ ) or father ( $pa$ ). Here a random population of chromosomes is shown in matrix  $A$ , where  $pops$  is total number of chromosomes,  $M$  is number of secondary users.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{pops1} & a_{pops2} & \cdots & a_{popsM} \end{bmatrix} \quad (9)$$

It starts by randomly selecting a variable in the first pair of parents to be the crossover point. In the figure,  $\alpha$  is the crossover point and  $\beta$  is a value randomly chosen in the range

of  $[0, 1]$ . As for the GA crossover operation, two parents are chosen and the new off springs are formed from combinations of these parents. For crossover scheme used in our proposed algorithm is a hybridization of an extrapolation method with a crossover method to enhance the quality of obtainable solutions [13]. The GA crossover operation is graphically explained in Fig. 2.

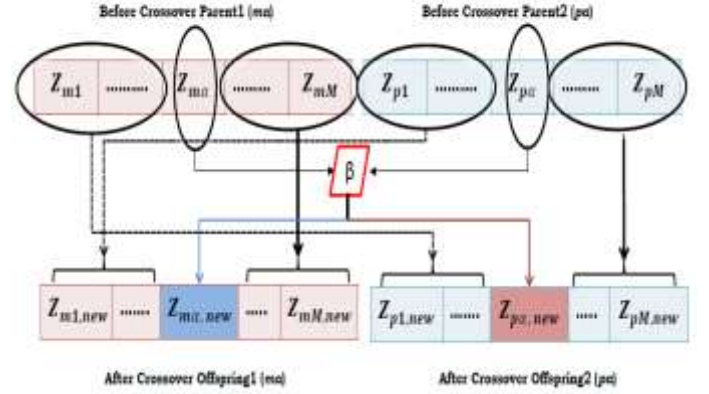


Figure 2. GA crossover operation.

For Parent1 ( $ma$ )  $\rightarrow$  offspring1 ( $ma$ ):

$$\begin{aligned} \{Z_{p1}, \dots, Z_{p(\alpha-1)}\} &\rightarrow \{Z_{m1, new}, \dots, Z_{m(\alpha-1), new}\} \\ \{Z_{m(\alpha+1)}, \dots, Z_{mM}\} &\rightarrow \{Z_{m(\alpha+1), new}, \dots, Z_{mM, new}\} \\ Z_{m\alpha, new} &= Z_{m\alpha} - \beta[Z_{m\alpha} - Z_{p\alpha}] \end{aligned} \quad (10)$$

For Parent2 ( $pa$ )  $\rightarrow$  offspring2 ( $pa$ ):

$$\begin{aligned} \{Z_{m1}, \dots, Z_{m(\alpha-1)}\} &\rightarrow \{Z_{p1, new}, \dots, Z_{p(\alpha-1), new}\} \\ \{Z_{p(\alpha+1)}, \dots, Z_{pM}\} &\rightarrow \{Z_{p(\alpha+1), new}, \dots, Z_{pM, new}\} \\ Z_{p\alpha, new} &= Z_{p\alpha} + \beta[Z_{m\alpha} - Z_{p\alpha}] \end{aligned} \quad (11)$$

The next step after crossover is the mutation operation. The total number of variables that can be mutated equals to the mutation rate times the population size. The row and column numbers of variables are nominated randomly and then these nominated variables are replaced by new random ones. For instance, if the mutation rate is 60% and the population size equal to 5 chromosomes as it shown in the matrix  $A$ , then the total number of variables that have to be mutated is  $0.6 * 5 = 3$  variables. Assume that the following pairs have been selected randomly from  $A$ :  $mrow = [4 \ 3 \ 5]$  and  $mcol = [2 \ 5 \ 1]$ , where  $mrow$  is the row index and  $mcol$  is the column index of the population. Then, the variables to be mutated, can be highlighted, as shown in the matrix  $A$  below.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \quad (12)$$

Assume that the 4<sup>th</sup> chromosome in  $A$  is defined as  $[a_{41} \ a_{42} \ a_{43}] = [0.0551 \ 0.8465 \ 0.9891 \ 0.2478 \ 0.0541]$ . Then, the mutation process of, for example, the variable  $A(4, 2) \rightarrow a_{42}$  is illustrated in Fig. 3. During the mutation operation, the previous value of  $a_{42} = 0.846$  is replaced by another random value and the new coefficient becomes  $a_{42} = 0.3041$ .

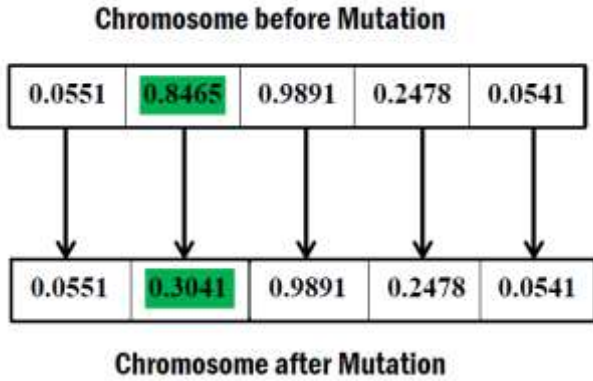


Figure 3. GA mutation representation.

### I. Mini-Max criteria for Genetic algorithm

The GA based optimization algorithm for SDF-based CSS can be outlined as follows:

**Step 1:** Set  $t = 0$  and randomly generate a population of  $pops$  chromosomes each of which is  $M$  digits long, where  $M$  is the number of secondary users in the network.

**Step 2:** Decode each chromosome in the random population into its corresponding weighting coefficients vector where the weighting coefficient vector  $\vec{\omega} = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]^T$ ;  $\omega_l > 0$  satisfying the condition; which is used to minimize the detection error.

**Step 3:** Normalize the weighting coefficient vector dividing  $\vec{\omega} = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]^T$  by its 2-norm such that  $\vec{\omega}_l = \vec{\omega}_l / \left[ \sum_{i=1}^M (\omega_i)^2 \right]^{1/2}$  so that the constraint  $\|\vec{\omega}\| = 1$  is satisfied.

**Step 4:** Compute the fitness value of every normalized decoded weighting vector,  $\vec{\omega}_i$  rank their corresponding chromosomes according to their fitness value and identify the best chromosomes  $\lfloor pops * elite \rfloor$ , where  $elite \in [0, 1]$   $\lfloor \cdot \rfloor$  denotes floor operation.

**Step 5:** Update  $t = t + 1$  and reproduce  $\lceil pops * (1 - elite) \rceil$  new chromosomes (candidate solutions) using GA operations: selection, crossover and mutation where  $\lceil \cdot \rceil$  denotes ceiling operation.

**Step 6:** Construct a new set of population  $pops$  by concatenating the newly  $\lceil pops * (1 - elite) \rceil$  reproduced chromosomes with the best  $\lfloor pops * elite \rfloor$  found in  $P(t - 1)$ .

**Step 7:** Decode and normalize the chromosomes of the new population  $pops$  as in Step 2 and Step 3 respectively.

**Step 8:** Evaluate the fitness value of each chromosome as in Step 4.

**Step 9:** If it is equal to the predefined number of generations (iterations)  $ngener$ , stop. Otherwise, go to Step 5.

## II. Neyman-Pearson criteria for Genetic algorithm

**Step 1:** Remains the same as Mini-Max criteria step.

**Step 2:** Remains the same as Mini-Max criteria step, but the condition must satisfy the condition: which is used to maximize the detection probability.

**Step 3:** Remains the same as Mini-Max criteria step.

**Step 4:** Remains the same as Mini-Max criteria step.

**Step 5:** Remains the same as Mini-Max criteria step.

**Step 6:** Remains the same as Mini-Max criteria step, but this time reproduced chromosomes with the best  $\lfloor pops * elite \rfloor$  found in  $P(t-1)$  (higher value).

**Step 7:** Remains the same as Mini-Max criteria.

**Step 8:** Remains the same as Mini-Max criteria.

**Step 9:** Remains the same as Mini-Max criteria.

### 2.2 PSO based cooperative spectrum sensing

PSO algorithm, invented by Kennedy and Eberhart in 1995 [14], is conceptualized from social performance of group of fishes and birds. The performance of these organizations is imitated by this amazing algorithm.



### I. Mini-Max criteria for particle swarm optimization algorithm

In this part, the problem is to minimize the objective function  $P_e(\vec{\omega})$  where  $\vec{\omega} = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]^T$  and  $M$  is the number of variables of  $P_e(W_2)$  with  $W' \in W \in W''$  where  $W' = 0$  and  $W' = 1$  are lower and upper limits on  $W$ . The steps involved in the PSO algorithm are as follows:

**Step 1:** The first step is the initialization of the algorithm and consists of randomly generating  $N$  numbers of particle position as  $\vec{\omega}_s^{(j)} = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]^T$  in the range of  $\omega^l$  and  $\omega^u$  and  $N$  numbers of length- $M$  particle velocity vector which are initially set to zero as  $\vec{v}_s^{(j)} = [0, 0, 0, \dots, 0]^T$ : ( $s = 1, 2, \dots, N$ ). To simplify the notation, particle position and velocity at iteration  $j$  are demonstrated by  $\vec{\omega}_s^{(j)}$  and  $\vec{v}_s^{(j)}$ , respectively.

**Step 2:** In this step, the value of the objective function for each of the particle positions generated in step 1 is calculated as  $P_e(\vec{\omega}_1^{(0)}), P_e(\vec{\omega}_2^{(0)}), \dots, P_e(\vec{\omega}_N^{(0)})$ .

**Step 3:** The objective functions' values achieved in step 2 are compared in this step and their smallest value is chosen. Next, the particle position equivalent to the minimum value is considered as  $P_{best,0}$  and iteration number is set to  $j = 1$ .

**Step 4:** The velocity of the  $s^{th}$  particle at the  $j^{th}$  iteration is updated based on the following equation:

$$\vec{v}_s^{(j)} = \vec{v}_s^{(j-1)} + c_1 r_1 \left[ P_{best,j} - \vec{\omega}_s^{(j-1)} \right] + c_2 r_2 \left[ G_{best,j} - \vec{\omega}_s^{(j-1)} \right] \quad (13)$$

where individual and social learning acceleration coefficients are, respectively, denoted by  $C_1$  and  $C_2$ ,  $r_1$ , and  $r_2 \sim U(0,1)$  are random numbers with uniform distributions in the range of 0 to 1 which introduce stochastic components to the algorithm. At the  $j^{th}$  iteration, the best experienced particle position which minimizes the objective function is denoted by  $P_{best+j}$ . The best qualified position among all iterations is called global best position and is expressed by  $G_{best}$ .

**Step 5:** At the iteration, the new position of the  $s^{th}$  particle is updated as follows:

$$\vec{\omega}_s^{(j)} = \vec{\omega}_s^{(j-1)} + \vec{v}_s^{(j)} \quad (14)$$

Again, the value of the objective function for each of the particle positions generated in this step is calculated as:

$$P_e(\vec{\omega}_1^{(j)}), P_e(\vec{\omega}_2^{(j)}), \dots, P_e(\vec{\omega}_N^{(j)}) \quad (15)$$

**Step 6:** In this step the comparison between the values of the objective functions obtained in step 5 is performed and the particle position corresponding to the minimum value of the objective function is defined as  $P_{best,j}$ . The value of the  $G_{best}$  will be replaced by the value of the  $P_{best,j}$  if the following condition is satisfied as follow:

$$Pe(P_{best,j}) \leq Pe(G_{best}) \quad (16)$$

**Step 7:** The convergence of the algorithm is checked in this step and if the algorithm converges to a stable value, the procedure is terminated. Otherwise, the iteration number is set to  $j = j + 1$  and the process is repeated from step 4.

### II. Neyman-Pearson criteria for particle swarm optimization algorithm

**Step 1:** Remains the same as Mini-Max criteria step.

**Step 2:** Evaluate the values of objective function corresponding to initial particle positions as  $P_d(\vec{\omega}_1^{(0)}), P_d(\vec{\omega}_2^{(0)}), \dots, P_d(\vec{\omega}_N^{(0)})$

**Step 3:** Find the maximum value of the objective function in the step 2 and set its corresponding particle position as the  $P_{best,0}$ . Set the iteration number  $j = 1$ .

**Step 4:** Like step 3 in Mini-Max criteria.

**Step 5:** Update the  $i^{th}$  particle position at the  $j^{th}$  iteration using:

$$\omega_i^{(j)} = \omega_i^{(j-1)} + v_i^{(j)}; (i = 1, \dots, N) \quad (17)$$

Evaluate the values of objective function corresponding to new particle positions as  $P_d(W_1^{(j)}), P_d(W_2^{(j)}), \dots, P_d(W_N^{(j)})$ .

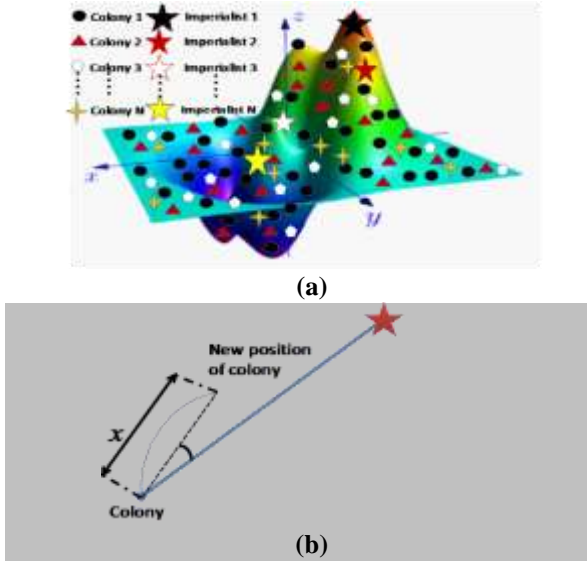
**Step 6:** Find the maximum value of the objective function in the step 5 and set its corresponding particle position as the  $P_{best,j}$ . If  $P_{best,j} \geq G_{best}$ , replace  $G_{best}$  with  $P_{best,j}$ .

**Step 7:** If the algorithm is converged to a stable value, stop the process. Otherwise, set the iteration number as  $j = j + 1$  and repeat from step 4.

### 2.3 Imperialistic Competitive Algorithm

ICA is inspired from imperialistic competition and human's socio-political evolutions [19-21]. The optimization targets in this research are respectively maximizing probability detection and minimize the probability of error in the Neyman-Pearson

and Mini-Max criteria. ICA begins with an initial population consisting of countries that are considered as individuals in other iterative based algorithms. This population is divided in two groups, a group with the finest (in Neyman-Pearson highest and in Mini-Max lowest) objective function values, power, which is chosen to be the imperialists, whereas the remaining group is their colonies. Then the colonies are distributed with the imperialists according to each imperialist power. Fig. 4 (a) depicts the initial colonies for each empire when superior empires have larger quantity of colonies e.g., imperialist 1. As an imperialist grows stronger, it will own more colonies. Empire is formed from an imperialist with its colonies in ICA language when each individual colony will try to discover better position to be named as imperialist of its empire. This method is fulfilled in ICA by moving the colonies towards their imperialist, named assimilation. It is possible that a colony turn out to be more powerful than its imperialist during assimilation, so the colony displaces the imperialist and the imperialist become one of its colonies. Moreover, it may be seen that through imperialistic competition the most powerful empires mind to raise their power, while weaker one's mind to break down. Both structures head the algorithm to steadily converge into a single empire, in which the imperialist and all the colonies have the similar culture.



**Figure 4.** a) Imperialists and colonies in each empire  
b) Movement of colony toward imperialist

### I. Neyman-Pearson criteria for imperialistic competitive algorithm

**Step 1:** The first step is the initialization of the algorithm. At the beginning Number of population ( $N_{pop}$ ) as  $\vec{\omega}_k = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]^T$  in the range of  $Var_{min}$  and

$Var_{max}$  is generated where  $N_{pop}$  is numbers of countries ( $k = 0 \dots N_{pop}$ ). The fitness value of each country is calculated and sorted, since Neyman-Pearson criteria need to use maximization algorithm, the imperialists are usually the countries with the highest objective function values. The amount of the objective function for each of the country generated in this step is calculated as:  $P_d(\vec{\omega}_1), P_d(\vec{\omega}_2), \dots, P_d(\vec{\omega}_{N_{pop}})$ .

**Step 2:** In this step, countries are divided into imperialists ( $N_{imp}$ ) and colonies ( $N_{col}$ ), we select  $N_{imp}$  from the most powerful countries. The colonies will be distributed among the imperialists according their power. This research proposes Boltzmann distribution [10] with suitable selection pressure coefficient ( $\alpha$ ).

$$P_{imp} \propto \exp(-\alpha P_d(\vec{\omega}_{imp})),$$

$$P_{imp} = \frac{\exp(-\alpha P_d(\vec{\omega}_{imp}))}{\sum_k \exp(-\alpha P_d(\vec{\omega}_k))} (k = 1, \dots, N_{pop}) \quad (18)$$

where ( $p_{imp}$ ) is probability of imperialist power ( $\sum_{N_{imp}} P_{imp} = 1$ ) from GA we already knew that optimum value for selection pressure ( $\alpha$ ) is when the sum of the half of the pest countries probability must be almost 0.8. The power of the imperialists is portion of  $N_{col}$  that should be possessed by  $N_{imp}$ . Number of colonies for each empire is randomly chosen from  $N_{col}$  in terms of power of its empire's imperialistic.

$$Initial \text{ number of empire's colonies} = rand[N_{col} \cdot P_{imp}] \quad (19)$$

**Step 3:** Every imperialist attempt to develop its colonies. Fig. 4 (b) depicts all colonies transfer to their related imperialist that  $x$  value is the colony transfers to. The new position of this colony is shown in darker blue. Value of  $x$  is uniformly chosen, i.e.  $x = U(0, \beta \times d)$ , where  $\beta$  is the assimilation coefficient ( $0 < \beta \leq 2$ ) and  $d$  is the distance of the colony and imperialist. In Fig. 4 (b), assimilation deviation ( $\theta$ ) is a uniform random distribution number which can be

chosen from  $-\frac{\Pi}{2} < \alpha \leq \frac{\Pi}{2}$ . In general, there is a trade-off

when we choose value of  $\theta, \beta$ , among the number of iterations, Exploitation and exploitation of the system. Section 4 describes the optimum values of assimilation parameters. We replaced the assimilation deviation with a random vector as follow to show the implementation of ICA:

$$\text{New position} = \text{old position} + U(0, \beta \times d) \otimes \text{rand}(V) \quad (20)$$

where base vector,  $V$  is beginning the former position of the colony and aiming to the imperialistic. Also, random vector and element-by-element multiplications are denoted by  $\text{rand}$  and  $\otimes$  sign respectively.

Inevitably, the colony is departed minus of consuming the classification of  $\theta$  because these random values are not same essentially. Stating a new vector in order to have suitable exploration (search area) capability satisfies the utilization of  $\theta$ .

**Step 4:** If a colony in empire has lower cost than imperialist the position of a colony and the imperialist will exchange. It means while colonies moving toward imperialist, one colony may rich to the better position (get more power) than imperialist.

**Step 5:** Calculate the total cost of all empires. Generally imperialist cost affects the cost of each empire, but to have an accurate view of an empire, the average cost of empire's colonies should not be eliminated. Below we have modeled this fact by stating the total cost of each empire:

$$(\text{Total Empire's cost})_n = \left( P_d(\vec{\omega}_{imp})_n + \xi \left( \frac{\sum_{j=1}^{N_{col}} P_d(\vec{\omega}_{col})}{N_{col}} \right) \right) \quad (21)$$

$$j = \{1, 2, \dots, N_{col}\} \quad n = \{1, 2, \dots, N_{imp}\}$$

where positive number  $\xi$  is less than one ( $0 < \xi \leq 1$ ). Slight value of  $\xi$  has less effect of empire's colonies on the whole cost of empire.

**Step 6:** Imperialistic competition. Choice the weakest colony from weakest empire and provide it to one of the best empires.

**Step 7:** Remove the defenseless empires. When all colonies of an empire move to other powerful empires and just imperialist remains, this imperialist automatically joins to best empires as a simple colony

**Step 8:** Stop condition will satisfy, if one empire remains. Otherwise, go to step 2. The result of the problem is the final Imperialist.

## II. Mini-Max criteria for modified imperialistic competitive algorithm

Steps of ICA for Mini-Max are very close to Neyman-Pearson criteria. Since Mini-Max is always used for minimization, some steps are different as below while applied parameters remains the same.

**Step 1:** The imperialists are usually the countries with the lowest objective function values. The amount of the objective function for each of the country generated in this step is calculated as  $P_e(\vec{\omega}_1), P_e(\vec{\omega}_2), \dots, P_e(\vec{\omega}_{N_{pop}})$ .

**Step 2:** The colonies will be distributed among the imperialists according their power:

$$P_{imp} \propto \exp(-\alpha P_e(\vec{\omega}_{imp})),$$

$$P_{imp} = \frac{\exp(-\alpha P_e(\vec{\omega}_{imp}))}{\sum_k \exp(-\alpha P_e(\vec{\omega}_k))} \quad (k = 1, \dots, N_{pop}) \quad (22)$$

**Step 4:** If a colony in empire has a higher cost than imperialist the position of a colony and the imperialist will exchange.

**Step 5:** The total cost of each empire:

$$(\text{Total Empire's cost})_n = \left( P_e(\vec{\omega}_{imp})_n + \xi \left( \frac{\sum_{j=1}^{N_{col}} P_e(\vec{\omega}_{col})}{N_{col}} \right) \right) \quad (23)$$

$$j = \{1, 2, \dots, N_{col}\} \quad n = \{1, 2, \dots, N_{imp}\}$$

## 3. Classification Results and Analysis

Table. 1 demonstrates the overall simulation parameters used in this paper. To realize the low SNR conditions (SNR < -10 dB) at SU and FC levels, the values of the  $g_i$  and  $h_i$  are generated randomly. In addition, since the channel is considered to be a slow fading channel,  $g_i$  and  $h_i$  values are assumed to be constant over the sensing time, so the delay requirement is short compared to the channel coherence time considered as quasi-static scenario [23]. Since the ICA, PSO and GA parameters are generally problem-dependent, the set-and-test approach is used in this work to obtain the optimal values for them as it is shown in Table. 2 and Table. 3. For instance  $c_1 r_1$  and  $c_2 r_2$  in PSO or  $\theta, \beta, \alpha$  in ICA guarantee that the particles or colonies would fly over the target about half the time [16].

### 4.1 Results and Analysis for Neyman-Pearson criteria

Fig. 5 illustrates the probability of detection of ICA-based scheme as well as all other conventional schemes in terms of the different probabilities of false alarm. It is observable that ICA-assisted method outperforms all other schemes with a large difference which validates the robustness of our proposed technique. For instance, for the fixed  $P_f = 0.1$ , the  $P_d$  provided by ICA is 97.8%, which is 0.9 %, 4.81%, 14.63%, 17.12% , 37.86% and 70.26% higher than PSO and GA respectively.

**Table 1.** Simulation parameters

CR and channel parameters	
Number of users $M$	20
Bandwidth $B$ , Sensing time $T_s$	6MHZ, 25 $\mu$ sec
SU transmit power $P_{R,i}$	12dbm
The step size of $P_f$ ( $0 < P_f < 1$ )	0.01
$P_{R,i}, \sigma_s^2$	33dbm, 35dbm
AWGN of $i^{th}$ PU-SU Channel	$20 \leq \sigma_{wi}^2 \leq 30$ dBm
AWGN of $i^{th}$ SU-FC Channel	$20 \leq \delta_i^2 \leq 30$ dBm
channel gains	$10 \leq g_i \leq 20$ dBm
channel gains	$10 \leq h_i \leq 20$ dBm

The convergence comparison of ICA-, PSO- and GA- assisted schemes over 200 iterations per simulation for a given  $P_f = 0.2$  is shown in Fig. 5. It is clearly shown that the Max ICA-based method converges roughly after the 29 iterations while the convergence for Max PSO- and GA-based method techniques are attained after 42 and 56 iterations respectively which implies the fast convergence of the ICA algorithm. The approximate improvement of 30.8% and 48.2% of the convergence speed of ICA-based method compare to PSO- and GA-based schemes confirms stability of the algorithm for real time applications. Also, the mean of each algorithm over 100 simulations is shown and compared with its maximum iteration in Table 4 which confirms that most of the discovered

weighting vectors in each simulation of ICA-based are better than PSO and GA-based.

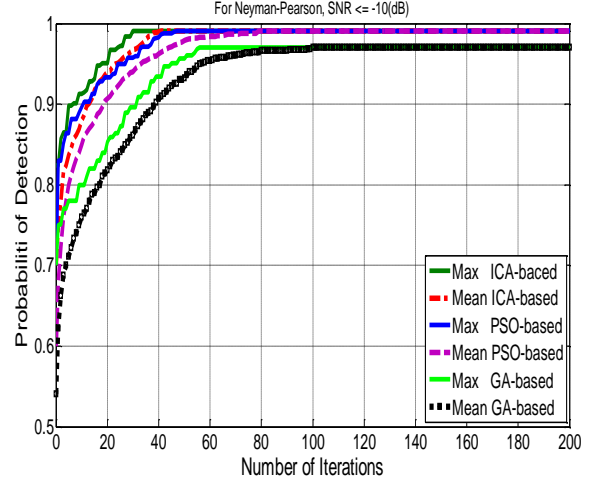
**Figure 5.** Comparison of probability of detection over 200 iterations for fixed  $P_f$  of 0.2

Table 4 shows Comparison of probability of error versus SNR for ICA- and PSO- assisted scheme which the effect of number of the population is quite visible. In general with increasing the number of population, performing time and computational complexity of these methods will be increased. On the other hand, choosing the number of countries and particles are problem dependent and depend on different factors like number of iterations, learning coefficient, assimilation coefficient, deviation coefficient and so on, e.g., the performance of the ICA with 25 countries is the best among all but in comparison with larger numbers of countries there is still a trade-off between very slightly improvements, complexity and performing time, which is well explained in Table 4.

**Table 2.** Different parameter values used for testing

GA		PSO		ICA	
Population size	10,20,30,40,50	Population size	5,10,15,20,25,50	Population size	5,10,15,20,25,35
Mutation rate	0.01,0.10,0.15,0.2, 0.3, 0.4,0.5,0.6	learning coefficients	1.8,1.85,1.9,1.95 ,2, 2.05,2.1	Mean colonies power coefficient	$0 < \xi \leq 1$
Crossover rate	0.5, 0.65, 0.75 , 0.85, 0.95	$r_1$ and $r_2$	U(0,1)	Assimilation coefficient	$0 < \beta \leq 2$
Population for reproduction rate	0.5 , 0.6 , 0.7 , 0.8 , 0.9	NON	NON	selection pressure	$0 < \alpha \leq 2$

**Table 3.** Optimal parameter values

GA		PSO		ICA	
Population size	50	Population size	25	Population size	25
Mutation rate	0.3	learning coefficients	2	Mean colonies power coefficient	0.15
Crossover rate	0.95	$r_1$ and $r_2$	NON	Assimilation coefficient	1.7
Population for reproduction rate	0.9	NON	NON	selection pressure	1.2



**Table 4.** Comparison of performance of ICA- and PSO-assisted for different number population.

ICA						PSO						
Number of Countries	5	10	15	20	25	Number of Particles	5	10	15	20	25	50
Probability of Max Detection ( $P_f = 0.2$ )	0.678	0.928	0.967	0.989	0.991	Probability of Max Detection ( $P_f = 0.2$ )	0.760	0.859	0.930	0.964	0.985	0.989
Max Convergence Iterations	195	128	74	48	24	Max Convergence Iterations	128	98	79	54	41	29
Mean Convergence iterations	Not	Not	101	56	38	Mean Convergence iterations	168	112	96	71	57	48
Max Iterations(Min)	0.0002	0.009	0.031	0.068	0.093	Max Iterations(Min)	0.089	0.136	0.182	0.236	0.275	0.562
Mean Iterations(Min)	Not	Not	1.847	5.758	9.270	Mean Iterations(Min)	8.702	13.405	18.282	24.047	26.571	56.538

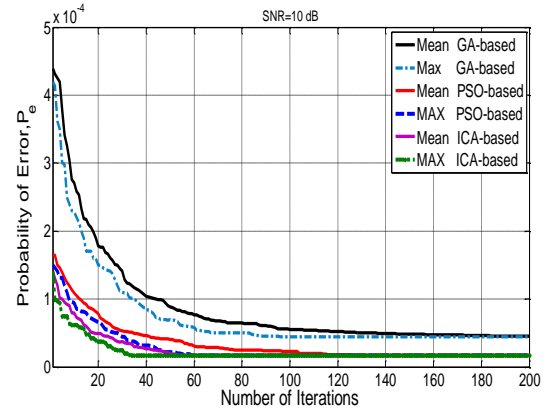
‘NON’ in Table 3 indicates that in each simulation ICA-based does not convergence within 200 iterations, in other words, ICA-based needs more iterations to be converged for the number of 5 and 10 populations when the number of imperialist is set to 5. Additionally, convergence time is computer-dependant and varies from one to another.

#### 4.2 Results and Analysis for Mini-Max criteria

Fig. 6 compares the convergence rate of ICA-assisted based with other schemes. Here, the probability of error over 200 iterations is evaluated for Iterative based methods in both conditions of Mean and Max. The mean and Max of each algorithm are achieved when the algorithms run for 100 times and the average of all results is called mean in our experiment and the best one among these all 100 simulations which results minimum  $P_e$  is named as Max algorithm-based. As it is seen, to achieve a probability of error equal to  $0.5 \times 10^{-4}$ , the mean ICA-based requires about 23 iterations whereas the same error rate can be obtained after 38 and 124 iterations for mean PSO- and GA-based, respectively. In addition, after the test duration of 200 iterations, the ICA and PSO algorithms are converging to the probability of error of about  $0.2 \times 10^{-4}$  while GA achieves  $0.45 \times 10^{-4}$  with the same number of iterations. The impact of different number of population and countries in CRN simulation is also examined for ICA- and PSO-based methods which are described in Fig. 8. It is apparent that improvement in performance is achieved by

increasing the number of population in each algorithm. It is notable that there is always a trade-off between performance and complexity in network when number of population increases.

Additionally, when the number of SUs in the system increases the cooperation with the system will be increased. As a result, separation between two hypotheses ( $H_0$  and  $H_1$ ) increases and the error performance of the system improves accordingly.

**Figure 6.** Comparison of probability of error over 200 iterations for ICA, PSO and GA**Table 5.** Comparison of performance of ICA- and PSO-assisted for different number population.

ICA						PSO						
Number of Countries	5	10	15	20	25	Number of Particles	5	10	15	20	25	50
Probability of Max Error ( $SNR = 5$ )	0.397 1	0.282 8	0.171 7	0.139 89	0.099 1	Probability of Max Error ( $SNR = 5$ )	0.430 5	0.35 32	0.231 8	0.186 4	0.1482	0.1009

Max Convergence Iterations	128	109	63	34	33	Max Convergence Iterations	117	98	79	67	59	38
Mean Convergence iterations	Not	189	131	72	53	Mean Convergence iterations	186	167	139	127	117	106
Max Iterations(Min)	0.000 2	0.009	0.031	0.068	0.093	Max Iterations(Min)	0.089	0.13 6	0.182	0.236	0.275	0.562
Mean Iterations(Min)	Not	1.821	4.141	4.891	5.201	Mean Iterations(Min)	17.02 8	23.1 34	26.93 1	32.73 9	35.842	62.284

## Conclusion

Cognitive radio cooperative spectrum sensing (CRCSS) is one of the interesting methods and well investigated to be known as a frequently monitoring mechanism which allows SUs to use detected spectrum holes caused by PU absence. A main challenge facing CSS in CR networks is selecting the appropriate weighting coefficient for each individual SU in fusion center (FC). Consequently, techniques to optimize these coefficients are fundamentals to the overall detection performance of the system. This research presents best existing methods in CRCSS according to previously achieved results by other researches, namely ICA, PSO- and GA-based methods in both criteria Neyman-Pearson and Mini-Max criterion for the first time. The achieved results confirmed that the ICA-based scheme outperforms all other SDF-based schemes with its less complexity which makes this method very close to real time application.

## References

- [1] Federal Communications Commission, "Spectrum policy task force report, FCC 02-155," Nov. 2002. doi:10.1109/DYSPAN.2007.88
- [2] S. Althunibat, M. Di Renzo, and F. Granelli, "Cooperative spectrum sensing for cognitive radio networks under limited time constraints," *Computer Communications*, vol. 43, pp. 55-63, 2014.
- [3] D. Cabric, A. Tkachenko, and R. W. Brodersen, "Spectrum sensing measurements of pilot, energy, and collaborative detection," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, 2006, pp. 1-7.
- [4] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 4, pp. 40-62, 2011.
- [5] S. Althunibat and F. Granelli, "An Objection-based Collaborative Spectrum Sensing for Cognitive Radio Networks," 2014.
- [6] X. Ling, Z. Bao, H. Wen, B. Wu, S. Xu, and L. Pan, "Cooperative spectrum sensing with adaptive energy threshold control and efficient data fusion," in *Communications in China (ICCC), 2013 IEEE/CIC International Conference on*, 2013, pp. 362-367.
- [7] W. Zhang, R. K. Mallik, and K. Letaief, "Cooperative spectrum sensing optimization in cognitive radio networks," in *Communications, 2008. ICC'08. IEEE International Conference on*, 2008, pp. 3411-3415.
- [8] E. Soltanmohammadi, M. Orooji, and M. Naraghi-Pour, "Improving the Sensing-Throughput Tradeoff for Cognitive Radios in Rayleigh Fading Channels," *Vehicular Technology, IEEE Transactions on*, vol. 62, pp. 2118-2130, 2013.
- [9] Z. Quan, S. Cui, and A. H. Sayed, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, pp. 28-40, 2008.
- [10] B. Shen and K. S. Kwak, "Soft combination schemes for cooperative spectrum sensing in cognitive radio networks," *ETRI journal*, vol. 31, pp. 263-270, 2009.
- [11] Sadeghi, H., Azmi, P., & Arezumand, H. (2012). Cyclostationarity-based soft cooperative spectrum sensing for cognitive radio networks. *Communications, IET*, 6(1), 29-38.. doi: 10.1049/iet-com.2011.0269.
- [12] A. A. El-Saleh, M. Ismail, M. A. M. Ali, and I. H. Arka, "Hybrid SDF-HDF cluster-based fusion scheme for cooperative spectrum sensing in cognitive radio networks," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 4, pp. 1023-1041, 2010.
- [13] S. Nazari-Shirkouhi, H. Eivazy, R. Ghodsi, K. Rezaie, and E. Atashpaz-Gargari, "Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm," *Expert Systems with Applications*, vol. 37, pp. 7615-7626, 2010.
- [14] O. Alink, A. B. Kokkeler, E. A. Klumperink, G. J. Smit, and B. Nauta, "Spectrum sensing with high sensitivity and interferer robustness using cross-correlation energy detection," 2013.
- [15] Neyman, J., & Pearson, E. S. (1933). On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231, 289-337. doi:10.1109/TSP.2003.809370..
- [16] M. Akbari and M. Ghanbarisabagh, "A Novel Evolutionary-Based Cooperative Spectrum Sensing Mechanism for Cognitive Radio Networks," *Wireless Personal Communications*, pp. 1-14, 2014.

- [17] M. Akbari, M. R. Manesh, A. A. El-Saleh, and M. Ismail, "Improved soft fusion-based cooperative spectrum sensing using particle swarm optimization," *IEICE Electronics Express*, vol. 9, pp. 436-442, 2012.
- [18] M. Akbari, M. K. Hossain, M. R. Manesh, A. A. El-Saleh, and A. M. Kareem, "Minimizing Sensing Decision Error in Cognitive Radio Networks using Evolutionary Algorithms," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 6, pp. 2037-2051, 2012.
- [19] C. Lucas, Z. Nasiri-Gheidari, and F. Tootoonchian, "Application of an imperialist competitive algorithm to the design of a linear induction motor," *Energy Conversion and Management*, vol. 51, pp. 1407-1411, 2010.
- [20] E. A. Gargari and C. Lucas, "Designing an optimal PID controller using Colonial Competitive Algorithm," *First Iranian Joint Congress on Intelligent and Fuzzy Systems*, 2007.
- [21] E. A. Gargari, F. Hashemzadeh, R. Rajabioun, and C. Lucas, "Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process," *International Journal of Intelligent Computing and Cybernetics*, vol. 1, pp. 337-355, 2008.
- [22] K. S. Cole, "A Theory of Surface Conductance at an Electrolyte-Solid Interface," *Journal of Applied Physics*, vol. 3, pp. 114-118, 1932.

